Appendix C: Theoretical Background

This manual describes how to specify the geometry, material properties, and boundary conditions required by THERM. Once those have been defined by the user, THERM automatically, meshes the cross section, performs the heat transfer analysis, runs an error estimation, refines the mesh if necessary and finally returns the converged solution. This appendix describes the technical algorithms behind each of these functions in detail.

AUTOMATIC MESH GENERATION

The first step in a finite-element analysis is the definition of a mesh. This mesh is made up of a finite number of non-overlapping subregions, known as elements, that cover the whole region under analysis. The manual creation of a well-conditioned finite-element mesh requires sophisticated knowledge of the solution method. This labor-intensive component of the finite-element analysis is accomplished for the user in THERM by the automatic meshing algorithm, Finite Quadtree.[5] The Finite Quadtree mesh generator is based on a spatial decomposition procedure that represents the domain of an object as a set of non-overlapping squares, referred to as quadrants, that are stored in a hierarchic tree. The object to be meshed is placed in a square universe that entirely encloses it. The square represents the "parent" quadrant; it is subdivided into four, "daughter", quadrants. The remainder of the tree is then defined in a recursive manner by subdividing the boundary and interior quadrants until all quadrants are at the level dictated by the geometry of the object. The subdivisions continue until each quadrant contains only one material. The mesh control parameter determines the minimum number of subdivisions preformed by Quadtree. The higher the mesh control parameter the smaller the maximum size of the finite element in the model. After the quadtree has been defined, all terminal quadrants are checked to ensure that only one tree level difference exists between them and their neighbors so that there is not a large size difference between adjacent finite elements in the final mesh. The final step in the algorithm is to convert the quadtree into a combination of well shaped quadrilaterals and triangles. For analysis purposes all elements are treated as quadrilaterals. A triangle is defined as a quadrilateral element in which two of the nodes are coincident. Once the mesh is created it is sent to the finite element solver.

FINITE ELEMENT SOLUTION

The finite element solver in THERM, Conrad, is derived from the public-domain computer programs TOPAZ2D and FACET[6,7,12]. The governing equation for two-dimensional heat conduction, under the assumption of constant physical properties, is derived from the general energy equation and is given by the partial differential equation shown in Eq. 1.

$$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+q_g=0 \qquad (1)$$

where,

q_g = internal heat generation

subject to the following set of boundary conditions:

$q_f = 0$ adiabatic boundary condition

$T = f(x, y)$ Temperature boundary condition

$q_f = \overline{q}$ known heat flux, $\overline{q}$, boundary condition

$q_c = h(T - T_\infty)$ convection / linearized radiation boundary condition,

$q_r = \varepsilon_i \sigma T_i^{\,4} - \alpha_i H_i$ radiation boundary condition.

For radiation boundary conditions, THERM assumes that radiating surface segments are gray and isothermal, so absorptance equals emittance, or $\alpha = \varepsilon$. **H** is irradiation on the surface. Equation 2 shows the value of **H** for a surface **i**.

$$H_i = \frac{1}{1 - \varepsilon_i}\left(B_i - \varepsilon_i \sigma T_i^{\,4}\right) \tag{2}$$

where,

B_i = radiosity of the surface "i"

B_j = radiosity of the surface "j" that views surface "i"

$$B_i = \varepsilon_i \sigma T_i^{\,4} + (1 - \varepsilon_i)\sum_{j=1}^{n} F_{ij} B_j \,. \tag{3}$$

Equation 3 represents a system of **n** linear algebraic equations which is solved for $\mathbf{B_i}$, and substituted in Eq. 2.

View factors are calculated using the "cross-string" rule which is illustrated in Figure C-1 and given by Eq. 4.

$$F_{ij} = \frac{r_{12} + r_{21} - (r_{11} + r_{22})}{2L_i} \tag{4}$$

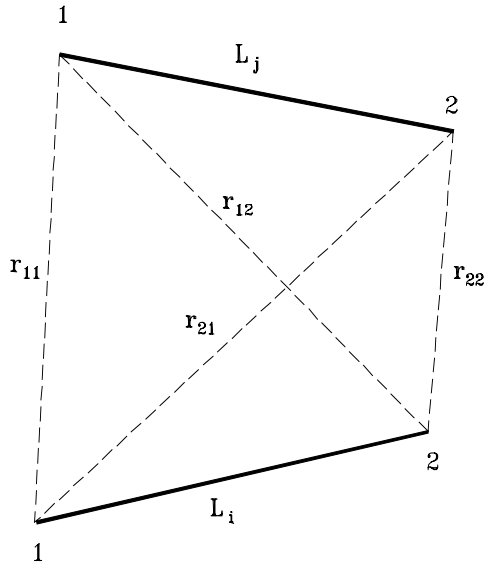

Figure C-1. Cross-String Rule

When partial, or third-surface shadowing exists, the two radiating surfaces are subdivided into **n** finite subsurfaces and contributions to the summation in Eq. 5 of those subsurfaces in which ray $\mathbf{r_{kl}}$ intersects a shadowing surface are excluded, as shown in Figure C-2.

$$F_{ij} = \sum_{k=1}^{n}\sum_{l=1}^{n} F_{kl} \tag{5}$$

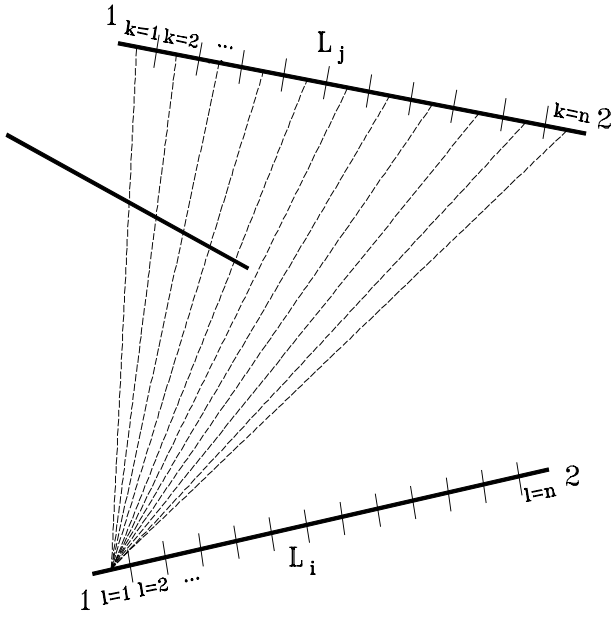

Figure C-2. Third-Surface Shadowing

The magnitude of the heat flux vector normal to the boundary, $\mathbf{q} = \mathbf{q_f} + \mathbf{q_c} + \mathbf{q}_r$ is given by Fourier's law, shown in Eq. 6.

$$q_f + q_c + q_r = -k\left(\frac{\partial T}{\partial x} n_x + \frac{\partial T}{\partial y} n_y\right) \tag{6}$$

THERM uses finite-element analysis based on the method of weighted residuals. The method of weighted residuals seeks to solve an equation of the form shown in Equation 7.

$$\iint_\Omega W\Re dxdy=0 \tag{7}$$

The residual function, $\Re$, results when an approximation for the actual temperature field is substituted in Eq. 1. The residual function is zero for the exact temperature field. The weighted residual method does not seek to force the residual function to zero everywhere in the domain; instead, the residual function is multiplied by a weighting factor, W, forcing the integral of the weighted expression to zero and minimizing the error of the approximate solution. This is accomplished by first integrating Eq. 6 by parts and substituting in the boundary conditions as shown in Eq. 8.

$$\begin{aligned} k\iint_\Omega\left(\frac{\partial W}{\partial x}\frac{\partial T}{\partial x} + \frac{\partial W}{\partial y}\frac{\partial T}{\partial y} - Wq_g\right)dx\,dy + \int_{\Gamma_h} Wh_c(T - T_\infty)\,d\Gamma_h - \int_{\Gamma_q} W\bar{q}\,d\Gamma_q \\ - \int_{\Gamma_r} W\left[\varepsilon_i\sigma\left(4\bar{T}_i^{\,3}T_i - 3\bar{T}_i^{\,4}\right) - \alpha_i H_i\right]d\Gamma_r = 0 \end{aligned} \tag{8}$$

Because the radiation boundary conditions make the problem non-linear, it is necessary to use an iterative technique to obtain the final solution. Term $\mathbf{T}_i^4$ is linearized by using the first two terms of its Taylor series expansion about $\bar{\mathbf{T}}_i$, the temperature from the previous iteration (the initial temperature is a guess for the first iteration).

Equation 8 is the "weak" form of the energy equation because it contains only first derivatives. The Galerkin form of the weighted residual method, used in THERM, relies on algebraic shape functions, $\mathbf{N}_i$, for the weighting function, $\mathbf{W} = \mathbf{N}_i$, and to approximate the temperature field, $\mathbf{T} = \mathbf{N}_i\mathbf{T}_i$. Equation 4 is set up for each element in a mesh. The integration over each element is approximated by a numerical integration using second-order Gaussian quadrature where the integration points are as shown in Figure C-3. The local systems of equations are combined into a global matrix, and a solution is obtained for the unknown nodal temperatures and heat fluxes.

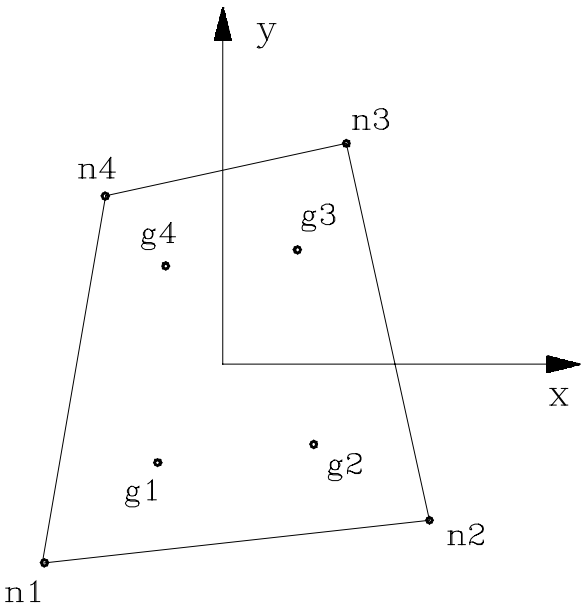

Figure C-3. Integration Points for Second-Order Gaussian Quadrature

Once the solution has been obtained on the initial mesh, it is sent to the error estimator, to check for convergence. The error estimator follows a published method [18,19]. Although a detailed description of this method is beyond the scope of this appendix we include a conceptual overview. Recall that the finite-element method solves the "weak" form of the energy equation, Eq. 4. We know that some points in the element have a higher order of convergence than others have, these are called superconvergent points. For quadrilateral elements used in CONRAD, the superconvergent points coincide with the Gaussian integration points, (g1-g4 in Figure C-3). The first step in error estimation is to use this knowledge of the finite-element solution to make a good guess at a better solution at all the nodal points. To accomplish this, a least squares method is used to fit a smooth function to the values of the gradient at the supercovergent points in a patch of elements around the node. The patch of elements around a node is defined as all the elements sharing that node. A better solution for the vector field of heat fluxes, $\mathbf{q_r}$, at the nodal points is recovered from this smooth function. The shape functions from the original problem are then used to interpolate the nodal solutions in order to obtain values for the recovered solutions throughout each element. Using the recovered fluxes, $\mathbf{q_r}$, and the original fluxes, $\mathbf{q}$, we calculate the contribution to the global error for each element as:

$$\|\mathbf{e}\| = \left(\int_{\Omega} (\mathbf{q_r} - \mathbf{q})^T (\mathbf{q_r} - \mathbf{q}) d\Omega \right)^{1/2} \quad (9)$$

If the global error is above a specified value, then the error estimator signals the mesh generator, and the mesh is refined in areas where the potential for error is high. The refined mesh is sent back to CONRAD, and a new solution is obtained. This process continues until the global error is less than the predetermined convergence value. The percent error energy norm is the maximum value of the error energy norm divided by the energy norm of the sum of the recovered fluxes and the error multiplied by 100:

$$\% \text{ Error Energy Norm} = \frac{\|\mathbf{e}\|}{\|\mathbf{q_r}\|} \times 100 \tag{10}$$

NOMENCLATURE

$\mathbf{e}$	error estimate
h	Surface heat transfer coefficient
k	Thermal conductivity
n_x, n_y	Vector components of the outward facing normal to the boundary
q_g	Internal energy source
$\bar{q}$	Known heat flux on the boundary
$\mathbf{q}$	Nodal flux vector used in error estimation
T	Temperature
x,y	Spatial coordinates
N	Shape functions
$\Re$	Residual function
W	Weighting function
Γ	Boundary surface
Ω	Element or problem domain.

Subscripts

∞	Ambient Conditions

Superscripts

T	Transpose